COLLECTION OF DATA ON MALFUNCTION AS PART OF COMPUTER RE- START

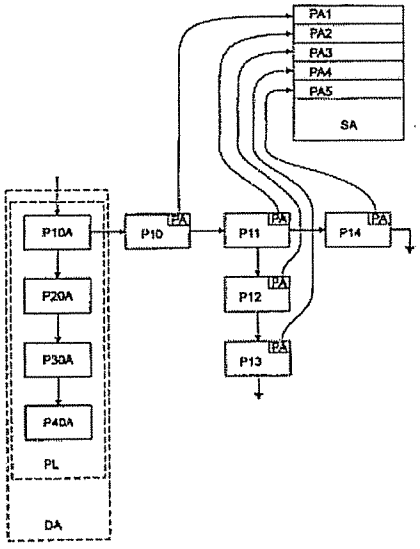
Publication number: RU2177636
Publication date: 2001-12-27
Inventor: TATTARI LAURI (FI)
Applicant: NOKIA TELECOMMUNICATIONS OY (FI)
Classification:
- international: ***G06F11/34; G06F11/00;*** **G06F11/34; G06F; G06F11/00;** (IPC1-7): G06F11/00
- European: G06F11/22A
Application number: RU19980110644 19961029
Priority number(s): FI19950005186 19951030

Also published as:
WO9716787 (A3)
WO9716787 (A2)
EP0870232 (A3)
EP0870232 (A2)
US6145095 (A1)
more >>

Report a data error here

Abstract of **RU2177636**

computer engineering. SUBSTANCE: procedure of collection of data on malfunction of computer consists in reading of structures of software data after initialization of re-start of computer from main storage of computer, in comparison of structures of data with normal values specified in advance prior to recording new data into part of main storage and in copying of structures of data read from main storage in peripheral determined beforehand in response to deviation. EFFECT: capability to collect data relevant to finding causes of errors and to store them for next analysis. 6 cl, 4 dwg



(19) **RU** (11) **2 177 636** (13) **C2**

(51) МПК[7] **G 06 F 11/00**

РОССИЙСКОЕ АГЕНТСТВО
ПО ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ РОССИЙСКОЙ ФЕДЕРАЦИИ

(21), (22) Заявка: 98110644/09, 29.10.1996

(24) Дата начала действия патента: 29.10.1996

(30) Приоритет: 30.10.1995 FI 955186

(43) Дата публикации заявки: 10.04.2000

(46) Дата публикации: 27.12.2001

(56) Ссылки: US 4381540 A, 26.04.1983. US 4930128 A, 29.05.1990. RU 2010313 C1, 30.03.1994. RU 2015523 C1, 30.06.1994. US 4593391 A, 03.06.1986. EP 0590866 A2, 30.02.1992.

(85) Дата перевода заявки PCT на национальную фазу: 01.06.1998

(86) Заявка PCT:
FI 96/00573 (29.10.1996)

(87) Публикация PCT:
WO 97/16787 (09.05.1997)

(98) Адрес для переписки:
129010, Москва, ул. Б. Спасская 25, стр.3, ООО "Юридическая фирма Городисский и Партнеры", Кузнецову Ю.Д.

(71) Заявитель:
НОКИА ТЕЛЕКОММЬЮНИКЕЙШНЗ ОЙ (FI)

(72) Изобретатель: ТАТТАРИ Лаури (FI)

(73) Патентообладатель:
НОКИА ТЕЛЕКОММЬЮНИКЕЙШНЗ ОЙ (FI)

(74) Патентный поверенный:
Кузнецов Юрий Дмитриевич

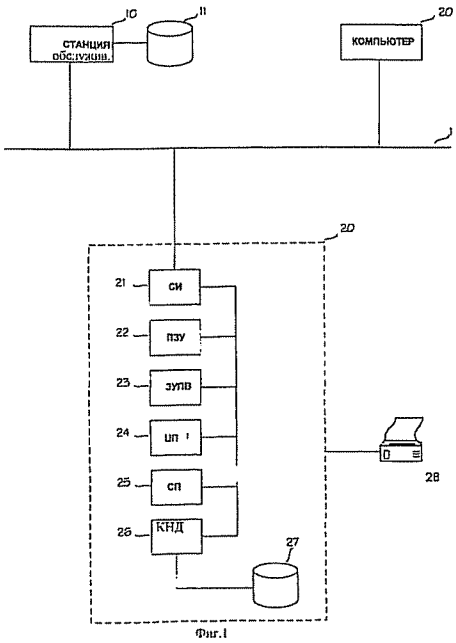
(54) СБОР ДАННЫХ ОБ ОТКАЗАХ КАК ЧАСТЬ ПЕРЕЗАПУСКА КОМПЬЮТЕРНОГО УСТРОЙСТВА

(57)

Изобретение относится к вычислительной технике. Техническим результатом является возможность сбора данных, являющихся релевантными при нахождении причин ошибки, и их сохранность для последующего анализа. Для этого заявленный способ состоит в том, что считывают из основного запоминающего устройства компьютера структуры данных программного обеспечения после инициализации перезапуска компьютера, но перед записью новых данных в часть основного запоминающего устройства, сравнивают содержимое структур данных с заранее определенными нормальными значениями и в ответ на отклонение копируют структуры данных, считанные из основного запоминающего устройства, в заранее определенное периферийное устройство. 6 з.п.ф-лы, 4 ил.

RU 2177636 C2

RU 2177636 C2



Фиг.1

RU 2177636 C2

RU 2177636 C2

(19) **RU** (11) **2 177 636** (13) **C2**

(51) Int. Cl.[7] **G 06 F 11/00**

RUSSIAN AGENCY
FOR PATENTS AND TRADEMARKS

(12) ABSTRACT OF INVENTION

(21), (22) Application: 98110644/09, 29.10.1996

(24) Effective date for property rights: 29.10.1996

(30) Priority: 30.10.1995 FI 955186

(43) Application published: 10.04.2000

(46) Date of publication: 27.12.2001

(85) Commencement of national phase: 01.06.1998

(86) PCT application:
FI 96/00573 (29.10.1996)

(87) PCT publication:
WO 97/16787 (09.05.1997)

(98) Mail address:
129010, Moskva, ul. B. Spasskaja 25, str.3,
OOO "Juridicheskaja firma Gorodisskij i
Partnery", Kuznetsovu Ju.D.

(71) Applicant:
NOKIA TELEKOMM'JuNIKEJShNZ OJ (FI)

(72) Inventor: TATTARI Lauri (FI)

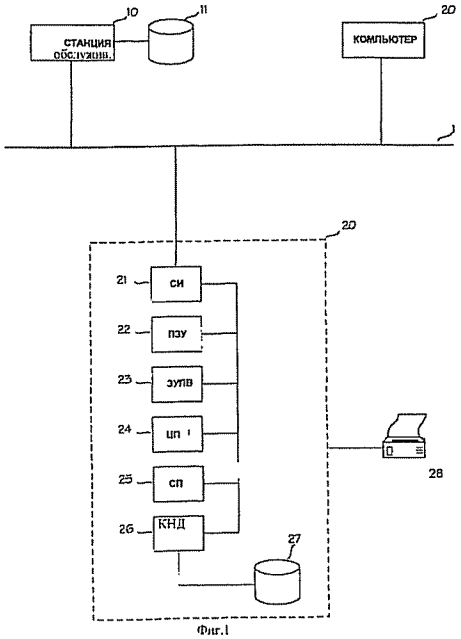
(73) Proprietor:
NOKIA TELEKOMM'JuNIKEJShNZ OJ (FI)

(74) Representative:
Kuznetsov Jurij Dmitrievich

(54) COLLECTION OF DATA ON MALFUNCTION AS PART OF COMPUTER RE- START

(57) Abstract:

FIELD: computer engineering. SUBSTANCE: procedure of collection of data on malfunction of computer consists in reading of structures of software data after initialization of re-start of computer from main storage of computer, in comparison of structures of data with normal values specified in advance prior to recording new data into part of main storage and in copying of structures of data read from main storage in peripheral determined beforehand in response to deviation. EFFECT: capability to collect data relevant to finding causes of errors and to store them for next analysis. 6 cl, 4 dwg



Фиг.1

RU 2177636 C2

RU 2177636 C2

Изобретение относится к сбору данных об отказах в компьютерных системах. Точнее, изобретение относится к способу решения проблем, ведущих к перезапуску компьютера, управляющего процессом, например - передачей вызовов.

Пользователи компьютеров и их программного обеспечения часто сталкиваются с такими беспокоящими ситуациями отказов, в которых компьютер переключается в исходное состояние, т.е. он действует так, как будто электропитание только что включено. В таком случае данные, которые хранились в запоминающем устройстве компьютера, обычно теряются. Операционная система может выдать короткое сообщение об ошибке, например - "Общий отказ защиты". Такое сообщение об ошибке вряд ли дает пользователю или поддерживающему консультанту какую-либо информацию о причине отказа или совет о том, как действовать, чтобы такая ситуация не повторилась. Другой экстремальный вариант представлен случаем, при котором программное обеспечение определенной локальной сети (Novell® Netware) сообщает о ситуации отказа. Когда ошибка возникает при выполнении функций программного обеспечения станции обслуживания сети, поддерживающий консультант получает возможность сохранить всю память станции обслуживания на дисках, которые могут понадобиться дюжинами. Очевидно, что непросто найти причину ошибки, если данных так много. Кроме того, проблема известных способов состоит в том, что при этих способах оповещение пользователя происходит до перезапуска компьютера. Если ситуация ошибки вызвала путаницу в записанных на дисках рабочих подпрограммах операционной системы, на диске нельзя сохранить ситуацию, предшествовавшую ошибке.

Перезапуски компьютеров происходят случайно. Для ясности отметим, что в настоящей заявке "перезапуск" относится, по существу, к перезапуску, причины которого следует проверить.

Очень важно, что в ситуациях ошибок операции технического обслуживания можно направлять правильно и по правильным местам. Например, в журнале "Формат ПК" (PC Format), август 1995 г., с. 27, в связи с отказами программного обеспечения компьютеров указано, что 15 января 1990 г. до половины телефонной сети "Американ Телефон энд Телеграф" (АТТ) в Соединенных Штатах вышла из строя и что в тот день 70 миллионов вызовов остались незавершенными. Отказ был отслежен, что привело к проведению операции технического обслуживания с программным обеспечением, управляющим телефонными станциями АТТ, причем эта работа не была выполнена, как планировалось.

На фиг. 1 показана распределенная компьютерная система. По меньшей мере, некоторые из компьютеров 10 и 20 (в этом примере компьютер 10) работают как станция обслуживания для других компьютеров 20. Станция обслуживания 10 содержит накопитель 11 на дисках, в котором хранится программное обеспечение системы. Компьютеры взаимосвязаны каналом 1, который может быть кабелем локальной вычислительной сети или дистанционно управляемым соединением, например - резервной модемной линией, соединением цифровой сети интегрального обслуживания (ЦСИО), радиолинией и т.п. Например, компьютер 20 запускается способом, который сам по себе известен, так что схемы поддержки (СП) 25 компьютера выдают сигнал запуска, который заставляет центральный процессор (ЦП) 24 переходить к заранее определенному адресу, содержащему информацию о постоянном запоминающем устройстве (ПЗУ) 22, которое содержит программу начальной загрузки. Выполнение программы начальной загрузки центральным процессором 24 предписывает, например, сетевому интерфейсу (СИ) 21 загрузить из станции обслуживания 10 через канал 1 сначала операционную систему, которая может содержать более усовершенствованные программы загрузки. Для выполнения этого операционная система и другие программы загружаются в ЗУПВ или запоминающее устройство 23 с произвольной выборкой, называемое далее основным запоминающим устройством. После загрузки операционной системы загружаются прикладные программы, и выполнение этих программ позволяет компьютеру 20 решать его реальную задачу. В качестве альтернативы загрузке программного обеспечения через канал 1, можно также загружать программное обеспечение через контроллер 26 накопителя на дисках (КНД) из локального накопителя 27 на дисках, если компьютер 20 содержит его. Компьютер 20 может также содержать устройство вывода 28.

На фиг. 2 показаны части, которые осуществлены для изобретения, в операционной системе компьютера 20. В этом, приведенном в качестве примера случае, операционная система ОС содержит четыре базовые функции ОС2-ОС5, т.е. управление процессами, управление памятью, передачу сообщений и исключение процессов.

На фиг. 3 изображена иллюстративная цепочка процесса. Точнее, эта цепочка является цепочкой блоков управления процессом (БУП). Эта цепочка может быть, например, цепочкой исполняемых процессов, цепочкой ожидания определенного события в семафоре, и т.п. Процессы $П_n$ обычно имеют сложную и динамичную структуру данных. На фиг. 3 цепочка процесса в целях иллюстрации упрощена таким образом, что структуры данных, связанные с каждым процессом $П_n$, сжаты в область параметров, ОП, которая содержит, например, данные о состоянии процесса и области памяти, зарезервированные для процесса, и данные о следующем и предыдущем процессе семейства. Когда процесс оканчивается, он должен освободить всю память, которую он зарезервировал. Общая причина проблем с памятью состоит в том, что процесс не освобождает всю память, которую он зарезервировал. Когда процесс резервирует память достаточное количество раз без освобождения памяти, которую он зарезервировал ранее, он, в конце концов, занимает всю имеющуюся память, так что другие процессы больше не могут резервировать память. Еще один механизм появления отказа может заключаться в том, что процесс ошибочно остается зацикленным и не принимает передаваемые ему

RU 2177636 C2

RU 2177636 C2

сообщения.

Формальное описание семафоров и операций приведено в работе И.В. Дийкстра "Последовательные процессы взаимодействия" в сборнике "Языки программирования" под ред. Ф. Гениуса, "Лондон Академик Пресс", 1965 ("Cooperating Sequential Processes" by E.W. Dijkstra in Programming Languages, ed. Genius, F., London Academic Press, 1965).

Вышеупомянутое описание процессов в значительной степени упрощено, но достаточно иллюстрирует проблему и ее решение. Если программа, загруженная в компьютер 20, а точнее - процесс программного обеспечения, приводит к отказу или, например, в компьютере возникает ошибка памяти, то компьютер перезапускается. В таком случае вышеупомянутая процедура загрузки программного обеспечения повторяется. Проблема локализации причин ошибки, которые ведут к перезапуску, состоит, прежде всего, в том, что трудно выделить релевантные данные из нерелевантных данных. Еще одна проблема состоит в том, что неправильная работа программного обеспечения может привести к нарушению осуществляемого программным обеспечением управления сетевым интерфейсом 21 и/или контроллером 26 накопителя на дисках, так что сохранение данных где-либо перед записью новых данных во время перезапуска может оказаться невозможным.

Сбор данных из некоторой области данных компьютера во время перезапуска уже известен. Например, в патенте США N 4930128 описан способ, при котором набор данных страницы сохраняют на жестком диске в качестве части начальной загрузки программы. Такое неизбирательное сохранение данных дает тому, кто пытается локализовать ошибку, слишком много нерелевантных данных. Как описано выше, такое огромное количество не выводимых на экран данных на практике не используется при локализации ошибок.

Исходя из вышеизложенного введения, задача изобретения состоит в том, чтобы разработать способ сбора данных об отказах компьютерных систем, при котором такие данные, которые могут быть релевантными при нахождении причин ошибки, сохраняются, например, для последующего анализа, а нерелевантные данные не сохраняются. Еще одна задача состоит в том, чтобы разработать аппаратуру для реализации этого способа.

Изобретение основано на том факте, что операционная система компьютера дополняется функцией, которая собирает во время перезапуска такие данные, которые могут оказаться важными при нахождении причины отказа. Согласно изобретению, области памяти компьютера инициализируются только тогда, когда сохранены части этих областей памяти, которые существенны для локализации отказа. Более точно, задачи изобретения решаются с помощью способов и аппаратуры, которые отличаются тем, что описано в независимых пунктах формулы изобретения. В зависимых пунктах формулы изобретения раскрыты различные пути более подробного определения сбора данных об отказах.

Способ и аппаратура, соответствующие изобретению, обеспечивают прежде всего то преимущество, что данные, связанные с перезапуском, можно заранее вводить на экран с помощью компьютера, так что нерелевантные данные не вызывают бесполезную загрузку ресурсов человека и компьютера. Минимальное использование ресурсов обеспечивает дополнительное преимущество, заключающееся в том, что данные можно также собирать даже с помощью аппаратуры, которая частично повреждена. Еще одно преимущество заключается в том, что данные сохраняются, когда компьютер находится в полностью определенном состоянии, так что в распоряжении имеются периферийные устройства и их драйверы, которые могут потребоваться для хранения данных.

Ниже изобретение описано более подробно в связи с предпочтительными конкретными вариантами осуществления и со ссылками на прилагаемые чертежи, на которых:

фиг. 1 изображает распределенную компьютерную систему,

фиг. 2 изображает части операционной системы компьютера,

фиг. 3 изображает иллюстративную цепочку процессов и

фиг. 4 является схемой, иллюстрирующей работу программы сбора данных об отказах.

Фиг. 2 изображает части операционной системы компьютера 20. Функция ОС1 сбора данных об отказах в соответствии с изобретением может быть помещена в операционную систему наряду с другими функциями операционной системы. На фиг. 2 функция ОС1 сбора данных об отказах показана как первая функция. Эта компоновка предназначена для того, чтобы подчеркнуть, что данные об отказах должны быть собраны до загрузки программного обеспечения, например - из накопителя на дисках, и повторной инициализации структур данных. На фиг. 2 также показано, что часть основного запоминающего устройства 23 компьютера 20 резервируется как область памяти, ОПАМ, использование которой будет пояснено в связи с фиг. 3.

Фиг. 3 изображает иллюстративную цепочку процессов. Программа ОС1 сбора данных об отказах, соответствующая изобретению, получает информацию о происхождении цепочки, например, посредством способов пакетирования программного обеспечения. Тот же способ используется, например, для разрешения использования библиотек общего назначения. Общая идея состоит в том, что программа ОС1 сбора данных об отказах, соответствующая изобретению, выявляет параметры различных процессов и семейств процессов таким же образом, как и остальное программное обеспечение - ОС2-ОС5 - выявляет соответствующие параметры. Начальную точку ОП10 процесса П10 локализуют на основе точки запуска. Процесс П10 связан с процессом П11, который, в свою очередь, связан с процессами П12 и П13. На фиг. 3 также показан в качестве примера случай, в котором процесс П11 может также быть связан с процессом П14. Предположим сначала, что процессы можно загрузить в запоминающее устройство динамически, т. е. по произвольным адресам, так что данные об

RU 2177636 C2

RU 2177636 C2

адресе запоминающего устройства, по которому помещен процесс, находятся, например, в области параметров предыдущего члена семейства.

Ниже описывается то, как программа сбора данных об отказах, соответствующая изобретению, может выявить, какой процесс остается зацикленным. Это зависание можно обнаружить, например, таким образом, что сообщение, которое было передано в процесс последним, сохраняется до тех пор, пока процесс не будет готов принять следующее сообщение. Если программа остается зацикленной, сообщение обычно в любом случае не выпускается. Программа ОС1 сбора данных об отказах, соответствующая изобретению, может обнаружить такую ситуацию, например, путем проверки блока управления процессом, БУП, в области параметров, ОП.

Кроме того, после нескольких процессов имеющаяся в наличии память фрагментируется таким образом, что расширенные процессы нельзя снабдить единой областью памяти. Такие проблемы можно решить, исходя из структур данных управления памятью операционной системы.

Вхождение в курс вышеупомянутых иллюстративных проблем облегчается, когда данные об областях памяти, зарезервированных процессами, сохраняют во время перезапуска следующим образом:

1) адрес ОП10 первого элемента П10 первого семейства процессов считывают из списка семейств процессов, ССП,

2) область параметров, ОП, первого процесса семейства процессов считывают на основе этого адреса и копируют в область памяти, ОПАМ,

3) этап 2) рекурсивно повторяют для всех процессов, которые вызвал этот процесс,

4) адрес первого элемента следующего семейства процессов считывают из списка и повторяют этапы 2) - 3) до тех пор, пока список семейств процессов не будет полностью обработан.

В этой связи отметим, что область параметров, ОП, относится к области в основном запоминающем устройстве 23, связанной с каждым процессом, и к сохранению данных, которые связаны с процессом и существенны для изобретения. Точный характер этих существенных данных будет описан ниже. Область памяти, ОПАМ, - это область, которая также находится в основном запоминающем устройстве 23, которая задается программным обеспечением сбора данных, соответствующим изобретению, и в которую области параметров, ОП, процессов копируются во время перезапуска.

Вышеупомянутый способ применим в общем случае, когда процессы можно загружать в любую область памяти. Когда определенные процессы всегда загружают по конкретному адресу, адрес начала и конца их областей параметров получают непосредственно известным способом во время компиляции программы. В случае программы анализа, соответствующей изобретению, эти адреса можно соотносить таким же образом, как их соотносят, когда задают также остаток программного обеспечения. В этом случае можно использовать более простой способ, который можно представить в виде следующего псевдокода:

```
копировать П10.ОП в ПАМ. ОП1
копировать П11.ОП в ОПАМ. ОП2
...
```

$П_n$. ОП относится к области параметров процесса $П_n$ и, соответственно, ОПАМ. $ОП_n$ относится к области в области памяти, ОПАМ, в которую копируется область параметров процесса $П_n$. В переводе на реальный язык программирования этот псевдокод, приведенный выше, можно реализовать таким образом, что адрес начала области П10. ОП загружается в исходный регистр в целях копирования, адрес начала области ОПАМ. ОП1 загружается в целевой регистр, а размер области П10. ОП в байтах загружается в регистр, указывающий количество байтов копирования. Адрес начала области ОПАМ. ОП2 получают путем прибавления размера области П10. ОП в байтах к адресу начала области ОПАМ. ОП1, и т.д. Соответственно, область параметров, ОП, любого процесса $П_n$ копируется в область памяти, ОПАМ, по адресу, который получается путем прибавления размера области параметров, ОП, предыдущего процесса $П_{n-1}$ к адресу начала области памяти, ОПАМ. $ОП_{n-1}$, предыдущего процесса.

Для иллюстрации изобретения, выше принято допущение, что область параметров, ОП, каждого процесса является единой областью памяти. Если область параметров распределена по нескольким отдельным областям, вышеупомянутые этапы копирования области параметров в область памяти, ОПАМ, следует повторять для каждого сегмента распределенной области параметров.

Значимые данные можно также получить путем проверки семафоров, относящихся к процессу. Семафор - это счетчик, который содержит состояние ожидания и посредством которого можно осуществлять взаимное исключение процессов. Процессы используют таким образом, что обычно только один процесс может оперировать с областью, защищенной семафором. Такая область, защищенная семафором, обрабатывается подпрограммой программного обеспечения, которая начинается с так называемой операции P и оканчивается так называемой операцией V. Операция P обеспечивает отрицательное приращение величины содержимого счетчика, и если эта величина слишком мала (отрицательна), процесс сам себя ставит в очередь в семафоре. В противном случае процесс продолжает код, следующий за операцией P. В конце кода процесс выполняет операцию V, при которой значение содержимого счетчика получает положительное приращение. Если значение стало отрицательным, активизируется процесс резервирования и он, в свою очередь, имеет доступ в защищенную область.

Проблема состоит в том, что когда процесс не попадает в защищенную область, т.е. когда он управляет семафором, очередь этого семафора прекращается без специальных мер. С другой стороны, программа может быть ошибочной и не выполнять операцию V вовсе. Эти проблемы можно решить путем указания данных обо всех семафорах, управляемых процессом, в области параметров.

Вышеописанный способ как таковой дает

RU 2177636 C2

RU 2177636 C2

слишком много недостаточно выводимых на экран данных для практических приложений. На самом деле данные следует выводить на экран более точно перед их передачей для анализа человеком. Например, можно поставить такую цель, чтобы данные, необходимые для устранения отказа, можно было сжать из основного запоминающего устройства объемом около 32 мегабайт в несколько или даже одну пригодную для чтения страницу. В таком случае из основного запоминающего устройства можно извлечь 0,01% данных, которые существенны для локализации отказа. Подходящий способ заключается в том, что программа сбора данных об отказах задает заранее определенное пороговое значение для памяти, резервируемой процессом. Сохраняются области параметров только тех процессов, которые зарезервировали память в количестве, превышающем это пороговое значение. Вместо этого программа сбора данных об отказах может также проходить по цепочке семейств процессов вышеуказанным образом, но дважды. Первый раз включает только выявление того, сколько памяти зарезервировано для каждого процесса или семейства процессов. Во второй раз в области памяти сохраняются области параметров нескольких или только одного семейства процессов, резервирующего больше всего памяти.

Для обобщения выше было внесено допущение, что список семейств процессов, ССП, является линейным списком, т.е. цепочкой, в которой за одним семейством процессов следует другое, но только одно семейство процессов (или конец списка). Также внесено допущение, что в рамках семейства процессов процессы $П_n$ могут образовывать древовидные структуры. Если реальная ситуация проще, чем одна из описанных выше, например - такова, что процессы образуют только линейные списки в рамках семейства процессов, обработку древовидных структур можно исключить для простоты.

Содержание областей параметров, ОП, существенное для изобретения, может включать в себя, например, следующие конкретные моменты процессов при использовании, например процессоров "Intell ®x 86":

- области памяти, зарезервированные процессом (размер, время резервирования, использование),
- сообщения, переданные, но еще не принятые процессом,
- управления временем, заданные процессом,
- файлы, открываемые процессом,
- совместно используемые ресурсы, зарезервированные процессом, например - операции P, выполняемые для совместно используемых семафоров,
- прикладные конкретные переменные в стеке и в области данных, а также данные, сохраняемые в этих областях операционной системой.

Соответственно, данные, совместно используемые всеми процессами, включают в себя:

- данные резервирования памяти, особенно - данные, относящиеся к имеющейся памяти,
- семафоры, связанные с резервированиями памяти,
- семафоры, предназначенные для использования в целях общего назначения,
- журнал регистрации ошибок операционной системы.

Программа ОС1 сбора данных об отказах, соответствующая изобретению, может находить эти структуры данных из памяти и получать данные из них так же, как остальная часть программного обеспечения обрабатывает те же структуры данных. Кроме того, выгодно дополнить программу сбора данных об отказах возможностью управления работой программы с принудительными управляющими записями, которые можно использовать, чтобы заставить программу собирать данные из определенных областей памяти. Эти области памяти, в которых осуществляется сбор, можно указывать посредством либо абсолютных адресов, либо символьных меток (если соответствующий перечень связей хранится в запоминающем устройстве компьютера). Поддерживающий консультант, пытающийся локализовать отказ, может формировать такие принудительные управляющие записи с помощью любой простой программы обработки слов. Эти записи могут содержать в простейшем виде только два параметра, адрес начала и адрес конца, и программа сбора данных об отказах копирует содержание области между этими адресами в область памяти. Программа может также копировать идентификатор, например - адрес начала. На практике отказы имеют тенденцию появляться в неожиданных местах. При наличии принудительных управляющих записей можно также собирать данные из мест, которые не предполагались при задании программы сбора данных об отказах.

Фиг. 4 изображает возможные этапы иллюстративного конкретного варианта программы ОС1 сбора данных об отказах, соответствующей изобретению. Программа сбора данных загружается в память вместе с остальной частью программного обеспечения (этап S0). На этом этапе переход, за которым происходит последующий перезапуск, задается так, что он направлен к программе сбора. Последующий перезапуск активизирует программу сбора (этап S5). Программа сбора считывает из основного запоминающего устройства 23 первую запись из списка принудительных управляющих записей (этап S10). Если такая запись существует, программа сбора обрабатывает ее (этап S20). Когда на этапе S15 обнаруживается, что достигнут конец списка принудительных управляющих записей, программа сбора переходит к этапу S25 для поиска структур данных процессов. На этапе S30 проверяют, все ли структуры данных обработаны. Если - не все, то на этапе S35 проверяют, содержит ли структура данных какие-либо аномальные признаки. Если это так, то на этапе S40 существенная часть структуры данных копируется в область памяти, ОПАМ. Когда на этапе S30 обнаружено, что достигнут конец списка структур данных, программа переходит к этапу S45 для загрузки других частей программного обеспечения, среди которых существенными для изобретения являются подпрограммы для обработки периферийных устройств, например - накопителей на дисках. Когда подпрограммы для обработки

RU 2177636 C2

RU 2177636 C2

периферийных устройств загружены, на этапе S50 содержимое области памяти, ОПАМ, копируется для использования поддерживающего консультанта. Поддерживающий консультант может получать содержимое области памяти, ОПАМ, в виде отчета, отпечатанного на принтере, в виде файла, записанного на диске, или сообщения по телефону на удаленную рабочую станцию.

В некоторых системах дефектная программа может полностью дезорганизовать основное запоминающее устройство, так что оно также вызывает неправильную работу программы сбора данных об отказе. Такой риск можно ограничить путем помещения программы сбора данных об отказах в запоминающее устройство, которое защищено, по крайней мере, от непреднамеренной перезаписи. Защиту можно осуществить, например, путем помещения программы сбора данных об отказах в ЗУПВ или ФЛЭШ-ЗУ или путем предотвращения записи в сегмент памяти, где размещена программа сбора, с помощью регистров процессора.

Конкретные варианты, иллюстрирующие изобретение, были проиллюстрированы на примере в связи с процессорами "Intell ®x 86". Очевидно, что изобретение этим не ограничивается, а может быть применено во всех типах процессоров таким образом, что, естественно, осуществляется сбор данных об управлении памятью и других данных, связанных с рассматриваемым семейством процессоров. Употребление терминов в данной области техники несколько варьируется. Термин "процесс", употребляемый в настоящей заявке, может быть назван "нитью" где-нибудь еще, так что термин "семейство процессов", употребляемый в настоящей заявке, должен был бы стать, соответственно, "процессом". Однако, объем изобретения не считается зависимым от употребляемых терминов. Кроме того, специалисту в данной области техники очевидно, что по мере развития технологии можно реализовать основную идею изобретения различными путями. Изобретение и конкретные варианты его осуществления не сводятся к вышеописанным примерам, а могут изменяться в рамках объема формулы изобретения.

Формула изобретения:

1. Способ сбора данных об отказах, относящихся к ситуациям ошибок, которые приводят к перезапуску компьютера, содержащего программное обеспечение с одним или несколькими процессами, отличающийся тем, что считывают из основного запоминающего устройства компьютера заранее определенное подмножество структур данных программного обеспечения после инициализации перезапуска компьютера, но перед записью новых данных в часть основного запоминающего устройства, где хранятся эти структуры данных, сравнивают содержимое структур данных с заранее определенными нормальными значениями и в ответ на отклонение содержимого указанных структур данных от заранее определенного нормального значения копируют структуры данных, считанные из основного запоминающего устройства, в заранее определенное периферийное устройство, чтобы поддержать содержимое структур данных после перезапуска.

2. Способ по п.1, отличающийся тем, что также перед перезапуском компьютера разервируют заранее определенную область памяти из основного запоминающего устройства компьютера для сбора данных об отказах и после перезапуска копируют структуры данных в заранее определенную область памяти перед их копированием в периферийное устройство.

3. Способ по п.1 или 2, отличающийся тем, что ситуацию ошибки, которая привела к перезапуску, относят к одному из ряда заранее определенных классов и лишь структуры данных, содержащие данные, относящиеся к рассматриваемому классу ситуации ошибок, копируют в периферийное устройство.

4. Способ по п.3, отличающийся тем, что классы ситуаций ошибок содержат по меньшей мере один из следующих классов: недостаточно памяти, задержание процесса в цикле, переполнение области памяти, выделенной для процесса, и неопределенная команда.

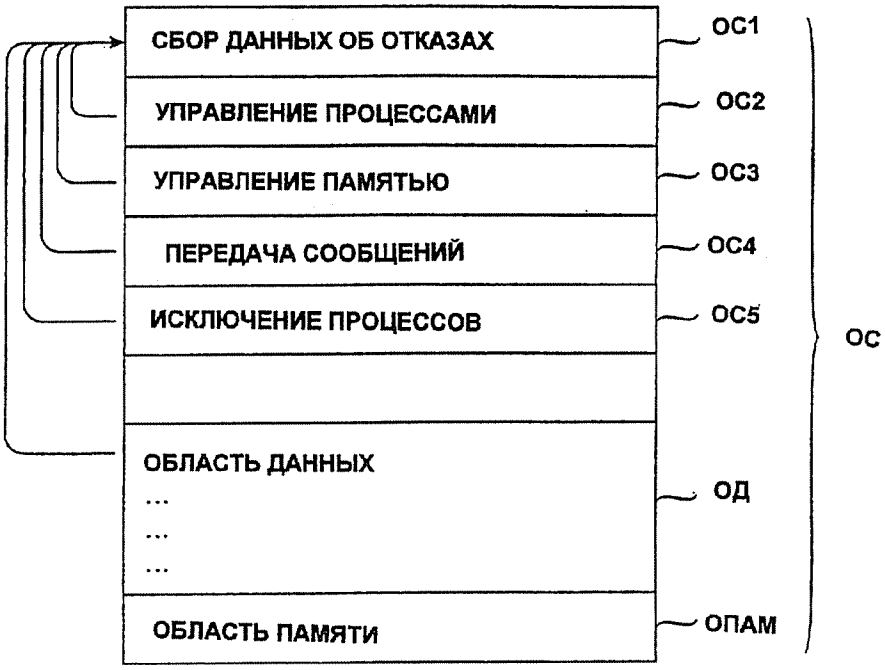
5. Способ по п.4, отличающийся тем, что когда ситуация ошибки принадлежит к классу "Недостаточно памяти", определяют один или самое большее несколько процессов, которые зарезервировали основное запоминающее устройство большей частью до перезапуска и копируют структуру данных в ответ на указанную структуру данных, содержащую данные, которые связаны самое большее с несколькими процессами, которые зарезервировали самую большую часть основного запоминающего устройства.

6. Способ по п.4, отличающийся тем, что если ситуация ошибки принадлежит к классу "Задержание процесса в цикле", определяют процессы, принявшие сообщения, которые процесс не выпустил, и копируют структуру данных в ответ на указанную структуру данных, содержащую данные, связанные с процессом, который не выпустил переданное ему сообщение.

7. Способ по любому из предыдущих пунктов, отличающийся тем, что считывают группу принудительных управляющих записей, содержащих структуры данных, и копируют структуры данных, содержащиеся в принудительных управляющих записях, в периферийное устройство.

RU 2177636 C2

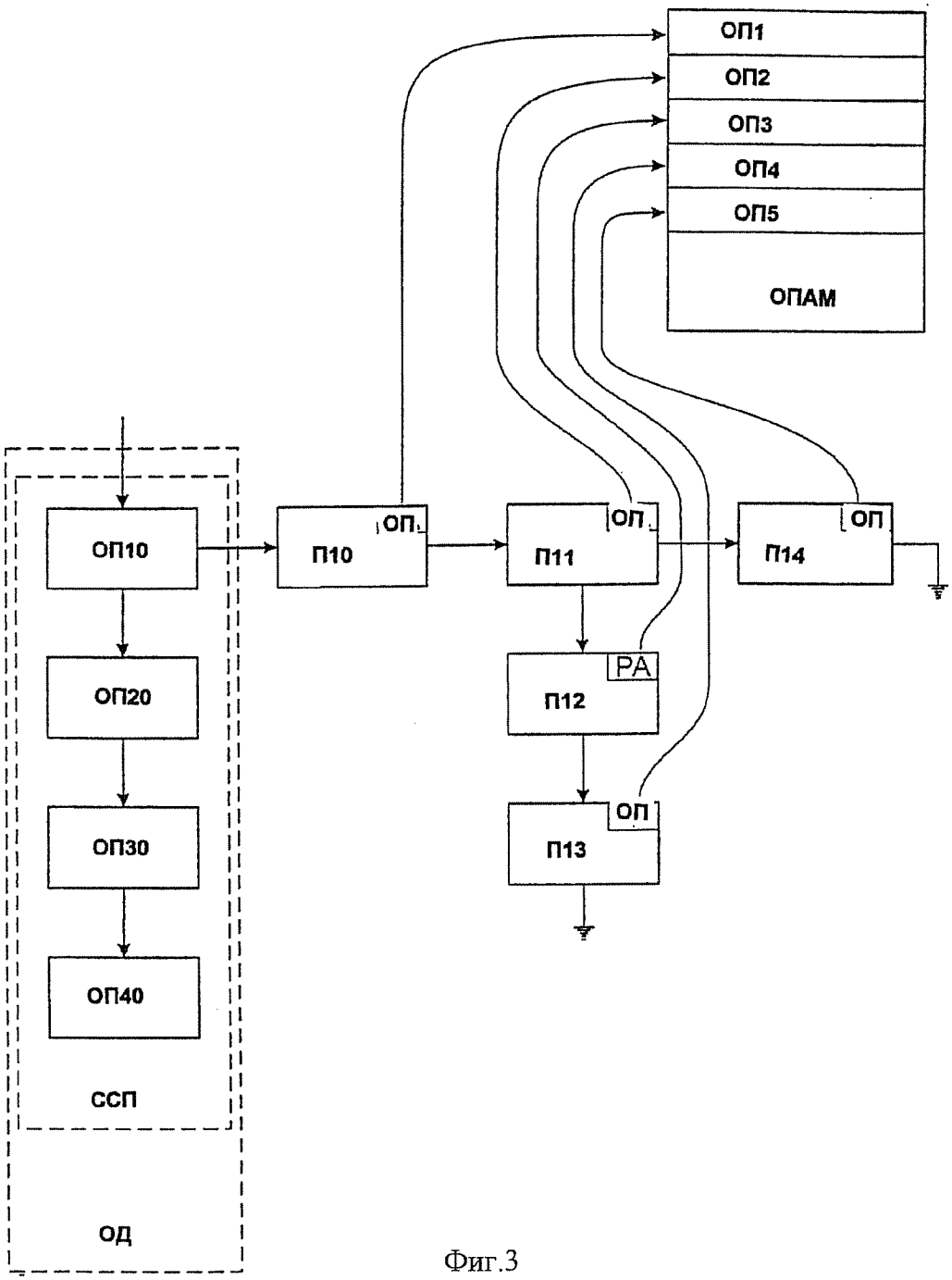
RU 2177636 C2



Фиг.2

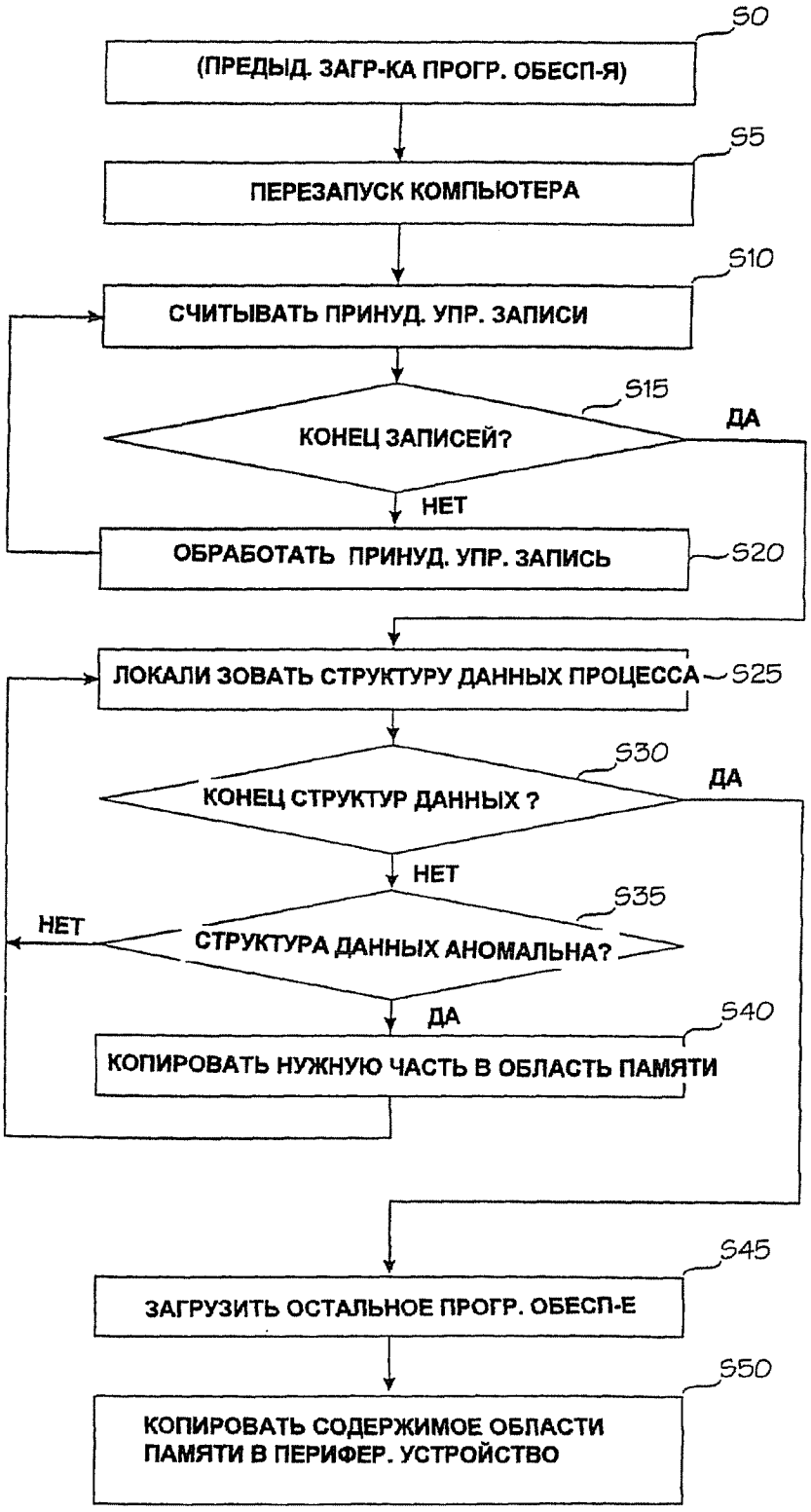
RU 2177636 C2

RU 2177636 C2



Фиг.3

RU 2177636 C2



Фиг.4

RU 2177636 C2

RU 2177636 C2